

77W AF\$



TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. RSW920010063US1
--	-------------------------------

Re Application Of: **Gabriel G. Montero, Srinivas Hati and David B. Colasurdo**

Application No. 09/823,120	Filing Date 3/30/2001	Examiner Avi M. Gold	Customer No.	Group Art Unit 2157	Confirmation No. 9260
-------------------------------	--------------------------	-------------------------	--------------	------------------------	--------------------------

Invention: **METHOD AND APPARATUS FOR ASYNCHRONOUS TIME BASED UPDATES OF HTTP SESSIONS**

COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on

The fee for filing this Appeal Brief is: **\$500.00**

- ☐ A check in the amount of the fee is enclosed.
- ☐ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **09-0461**.
- ☐ Payment by credit card. Form PTO-2038 is attached.

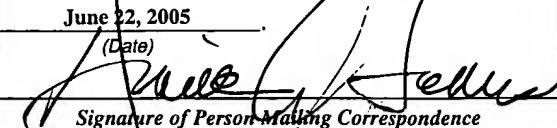
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.



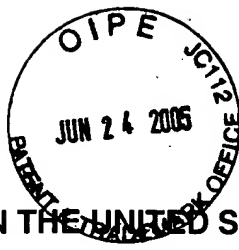
Signature

Dated: **6.22.05**

Theodore Naccarella, Esquire
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on June 22, 2005 (Date)	
 Signature of Person Mailing Correspondence	
Diane A. Sears Typed or Printed Name of Person Mailing Correspondence	

cc:



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor: **Gabriel G. Montero, Srinivas Hasti and David B. Colasurdo**

Application No.: **09/823,120**

Examiner: **Avi M. Gold**

Filed: **March 30, 2001**

Group Art Unit: **2157**

Case: **RSW9-2001-0063-US1**

Title: **METHOD AND APPARATUS FOR ASYNCHRONOUS TIME BASED
UPDATES OF HTTP SESSIONS**

CERTIFICATE OF MAILING/FACSIMILE

I hereby certify that this correspondence, along with any paper indicated as being enclosed, are being deposited with the United States Postal Service as first-class mail, postage pre-paid, on June 22, 2005 in an envelope addressed to Mail Stop Appeal Brief, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

22 June 2005
Date

Diane A. Sears
Diane A. Sears

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attention: **Board of Patent Appeals and Interferences**

APPELLANTS' BRIEF

This brief is in furtherance of the Notice of Appeal filed in this case on or about June 9, 2005.

1) REQUIRED FEE

The requisite fee of \$500.00 set forth in §41.20(b)(2) is submitted herewith. If the submitted fee is insufficient, the United States Patent and Trademark Office (hereinafter "Office") is authorized to charged Applicant's Deposit Account No. 19-5425 for any shortfall.

06/27/2005 MAHME1 00000011 090461 09823120

01 FC:1402 500.00 DA

2) REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation. Accordingly, International Business Machines Corporation is the real party in interest.

3) RELATED APPEALS AND INTERFERENCES

The appellant, assignee, and the legal representatives of both are unaware of any other appeal or interference that will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

4) STATUS OF CLAIMS

- a. Claims canceled: None
- b. Claims withdrawn from consideration but not canceled: None
- c. Claims pending: 1-24
- d. Claims allowed: None
- e. Claims merely objected to: None
- f. Claims rejected: 1-24
- g. Appealed Claims: 1-24

Appealed claims 1-24 as currently pending are attached as Appendix A hereto.

5) STATUS OF AMENDMENTS

In the response to the Final Office Action in this case, Applicant submitted a proposed amendment to the drawings to correct clerical errors in Figure 2 (namely, missing "YES" and "NO" labels on the lines exiting decision box 237 in the flowchart of Figure 2 and a missing connecting line out of decision box 237 corresponding to the "NO" branch). The Advisory Action did not indicate whether the proposed amendment to Figure 2 would or would not be entered. Accordingly, Applicant assumes those amendments have been approved for entry.

6) SUMMARY OF CLAIMED SUBJECT MATTER

The invention claimed in the claims under appeal is best illustrated by Figure 2 of the present application and is described in detail at page 12, line 10 through page 19, line 6 of the specification. The subject matter of all claims pertains to a method and apparatus for updating a session database that is accessible by multiple servers in a Web environment. Page 8, line 14-15. For instance, Web sites often divide the tasks of servicing requests into a three tier system with a different server or plurality of servers to handle each tier. The first, front end tier may be the http server that processes the http aspects of a transaction. The second tier may be the application servers, which handle the content specific processing for the transactions. The third tier may comprise database servers. Within each tier, the Web site server system may have multiple, redundant servers. Since http is a connectionless protocol, one request from a particular client can be directed to one application server while the next request from the

same client machine might be directed to a different application server. Accordingly, a means must be provided for the various servers to access the session data developed by another, redundant server.

Commonly, such sharing of http session data is enabled by use of a database server that is accessible to the plurality of application servers for storing session data. Particularly, an application server stores session data in local memory, but also writes a copy of the session data to the session database. If a different server services a request from a client, that different server can go to the database and read the session data for the corresponding session. The session data is updated in both the local memory and the database each time a request causes a change in the data.

The present invention is designed to reduce the number of writes to an http session database in order to conserve system resources. Page 12, lines 10-11. In accordance with the invention, each server maintains http session data in a local memory. This copy of the http session data will be updated every time there is a change in the session data. Furthermore, the server writes a copy of the session data to the common database shared by all of the servers only at designated times. In a preferred embodiment of the invention, the designated time is periodic. In alternate embodiments, the servers may write the session data to the database (a) after a specified number of requests in that session have been received or (b) after a specified number of changes to the session data have been made. Page 12, line 21 through page 13, line 16 and page 16, lines 8-17.

7) GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

1. Claims 11-13, and 16 are rejected under 35 USC section 103 (a) as unpatentable over U.S. Patent No. 6,539,494 issued to Abramson et al. (hereinafter Abramson) in view of U.S. Patent No. 6,178,439 issued to Feit (hereinafter Feit).

2. Claims 14 and 17 stand rejected under 35 U.S.C. §103(a) as unpatentable over Abramson and Feit further in view of U.S. Patent No. 6,701,438 issued to Prabandham. et al. (hereinafter Prabandham).

3. Claims 1-10, 15, and 18-24 stand rejected under 35 U.S.C. §103(a) as unpatentable over Abramson, Feit and Prabandham further in view of U.S. Patent No. 6,745,387 issued to Ng et al. (hereinafter Ng).

8) ARGUMENT**A. Claims 11-13 and 16**

The Examiner rejected claims 11-13 and 16 under 35 USC section 103 (a) as unpatentable over Abramson in view of Feit. In particular, the Examiner asserted that Abramson teaches almost all of the limitations of independent claim 11, including the limitation of “writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server systems at designated times”. The Examiner conceded that Abramson does not teach that the designated times are a function of a predetermined time interval since a last write to the database of data for said sessions. However, the Examiner further asserted that Feit teaches session control for HTTP communications over the Internet and the use of time

intervals. The Examiner, therefore, concluded that it would have been obvious to modify Abramson in view of Feit to use predetermined time intervals because it would avoid writing a copy of the data too often.

Applicant respectfully traverses. With respect to Abramson, the Examiner asserted that Abramson column 4, lines 5-16 teaches writing the session data to central memory at designated times. Overall, Abramson teaches nothing significantly different from what is described as prior art in the background section of the present application. Column 4, lines 5-16 of Abramson merely describe the type of data that is maintained as session data. This portion of Abramson is not even the portion that discusses writing that session data to the back-up server. There is no mention in this portion of Abramson of when or how that session data is written to the back-up server. There is nothing in Abramson that suggests that Abramson is doing anything different from what is described in the background section of the present application in connection with how and when it writes the session data to the back-up server. Applicant had previously argued during the prosecution of this case that the Examiner misunderstood Abramson. However, in view of the Final Office Action and the Advisory Action, it now appears that the Examiner might simply be giving the claim language a broad interpretation in which the recitation that writing to the database occurs “at designated times” encompasses any and all writing to a database because whatever is written to the database obviously is written at some point in time that can be called a “designated” time.

In any event, the rejection must fail because the prior art of record does not teach that the “designated times” are “a function of a predetermined time interval since a last write to said database of data for said sessions” as claimed.

Particularly, the rejection relies on the assertions that (1) Feit teaches this feature and (2) it is obvious to combine this alleged teaching of Feit with the system of Abramson. However, neither of these assertions is defensible.

Although Feit is the secondary reference in all of the rejections, Feit is the more significant reference in this analysis. Particularly, as previously noted, Abramson largely discloses that which Applicant has already admitted in the Background Of The Invention section of this application to be prior art. The Examiner does not appear to be asserting that Abramson teaches anything more. The present invention resides largely in when session data is written back to the database. The Examiner is relying exclusively on Feit with respect to this feature.

MPEP §2143 lists three requirements for a proper rejection based on obviousness, namely:

First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

The first and third requirements are lacking in the present case, i.e., (a) some suggestion or motivation to combine the referenced teachings and (b) the cited art teaching or suggesting all of the claim limitations. Particularly, the Examiner relies on the abstract of Feit and column 6, lines 7-13. Particularly, the Examiner asserts that

Feit's abstract teaches a session control for http communications over the Internet and that column 6, lines 7-13 teaches the use of time intervals. The Examiner then asserts that it would have been obvious to modify Abramson in view of Feit to use predetermined time intervals and designated times because it avoids writing a copy of the data too often and interfering with other processes in the network.

However, Feit's use of time intervals has absolutely nothing to do with the sessions discussed in Abramson or the present invention and therefore (1) does not teach anything about when to write session data to a common database and (2) does not provide any impetus to a skilled artisan to make the change to Abramson that would be necessary to arrive at the present invention.

In fact, while Feit does generally relate to http session control, it is apparent that Feit was developed prior to the existence of the type of Java session or session data that is the subject of the present invention. Specifically, Feit discusses at great length that there is no technology available for the Internet for maintaining anything like the session data that is the subject of the present invention. (See col. 1, line 65 – col. 2, line 51 of Feit). Rather, in Feit's terminology, an Internet "session" comprises a single http request and response (col. 2, lines 15-17).

The problem that Feit addresses is the desire to know when a true session (as that term is used in the present application, e.g., a continuous series of exchanges between a single client and a single server) is terminated. Apparently, at the time the technology described in Feit was developed, there was no true session data on the Internet, since, if there was, there would be no need for Feit's invention because Java http session data includes exactly this information.

In any event, the disclosure in Feit that the Examiner is relying upon is the “heart beat” feature. Particularly, when a client requests a page from a server, the server inserts extra code into the page before delivering it. This extra code causes the client to send a heart beat back to the server at defined time intervals. In response to receipt of the heart beat, the server sends a message back to the client acknowledging the heart beat. This continues until the client closes the page. The server keeps a record of the number of heart beats it receives from that page. When the server realizes that it has not received an expected “heart beat”, it assumes the session has been closed. It then determines the approximate length of the session by counting the number of heart beats it received from the page.

This has absolutely nothing to do with the present invention.

The claim recitation at issue in claim 11 is “writ[ing] to said data base, a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session”. Previous to this portion of the claim, the claim recites that the database is “accessible by each of the JVMs”

Thus, in order to reject this claim, at a minimum, there must be some teaching in the prior art of writing HttpSession data to a shared database at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session.

There is absolutely nothing in Abramson or Feit that teaches or suggests such a feature. Specifically, the Examiner essentially has conceded that Abramson does not teach it. Feit clearly does not teach this.

Feit's heart beat is sent from the client to the front end server, not from the local server to the back end database server. It does not have anything to do with writing to a database, let alone to a database that is shared among a plurality of JVMs, as claimed.

Feit's heartbeat has nothing to do with HttpSession data. The heart beat is nothing more than a ping from the client to the front-end server.

Feit's heartbeat has nothing to do with copying data to a database. The local server simply receives and counts up the heart beats.

The Examiner might as well have cited an alarm clock instead of Feit since the only relevant teaching in Feit that the Examiner appears to be relying on is that it does something (anything) at predetermined intervals. It cannot reasonably be asserted that any teaching of doing anything at predetermined intervals can be combined with a reference teaching writing a copy of HttpSession data to a database using a completely different schedule suggests the combination of writing the HttpSession data to the database at predetermined intervals. This turns the obviousness inquiry on its head.

Furthermore, the Examiner seems to assert that the sending of any piece of data by any computer constitutes writing to a database. This just is not a reasonable interpretation, particularly where the claim recites that the database is shared by multiple JVMs. Thus, even if one took this specious interpretation of the word "database", the rejection still fails because this "database" is not shared by multiple JVMs, as claimed.

It cannot rationally be said that a person of ordinary skill in the art having knowledge of Abramson would be motivated by Feit to change the timing of the copying

of HttpSession data from the local cache to the back-end database. Feit's heart beat has nothing to do with when to copy HttpSession data from a local cache to a back end database.

In order to arrive at the present invention, this ordinarily skilled artisan would need to discard every teaching of Feit except the teaching of using a fixed interval to do something totally different and substitute that concept into a very specific portion of Abramson that has absolutely nothing to do with Feit's heart beat. It should be clear that this is a not a realistic scenario.

The combination of the teachings of Abramson and Feit that would be necessary to arrive at the present invention simply is not suggested in the prior art. The Examiner has clearly used impermissible hindsight in order to pick and choose features from two different references without any rationale provided by the prior art for doing so.

Referring to independent claim 11, Abramson in combination with Feit does not teach or suggest "writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions".

Since claims 12, 13, and 16 depend from claim 11, they distinguished over the prior art for at least the same reasons.

B. Claims 14 and 17

Claims 14 and 17 stand rejected as obvious over Abramson and Feit further in view of Prabandham. Prabandham teaches nothing that was lacking from Abramson

and Feit as discussed above in connection with claim 11, from which claims 14 and 17 depend. Accordingly, claims 14 and 17 also distinguish over the prior art of record for at least the reasons set forth above in connection with claim 11.

Furthermore, claim 17 depends from claim 14 and adds the step of “polling said session objects stored in said memories local to said JVMs to determine if they have been updated since the last time step (2) was performed” and that “only copies of said HttpSession objects that have been updated within said predetermined time interval are written to said database”. The Examiner asserted that this is taught in col. 3, lines 24-30 and column 6, lines 50-67 of Abramson.

However, column 3, lines 24-30 discuss load distribution among servers and has nothing to do with session data. Column 6, lines 50-67 discusses the writing of session data to the backup server, but does not appear to disclose anything remotely resembling polling of application servers or selectively writing HttpSession objects.

The Examiner responded to this argument in the Final Office Action. Specifically, the Examiner asserted that column 3, lines 24-30 of Abramson disclose polling of each application server for status, which also inherently includes session data.

The underlined portion of the Examiner’s assertion is the key to this rejection since claim 17 recites “polling said session objects stored in said memory local to said JVMs to determine if they have been updated since the last time step (2) was performed”. The Examiner’s assertion that the data described as being polled in Abramson inherently includes session data is demonstrably false. Particular, as previously noted, this portion of Abramson discusses load distribution among servers and has nothing whatsoever to do with session data. Accordingly, there is absolutely no

reason to believe that the status information for each application server that Abramson is polling inherently includes session data and every reason to believe that it does not include such information. However, one need not rely on inference to determine that the Examiner's assertion is not true.

The cited portion of Abramson discloses that the status and load information that is polled is shown in Figure 2. A review of Figure 2 shows that, in fact, there is no session data. Instead, there is simply a status column (indicating that the status of the server is either "okay" or "not okay") and a probability column, which:

provides a set of probabilities, based on the status and load information, as to whether to select each application server 24. A more heavily loaded application server receives a lower probability (for example, Application Server No. 3, shown in FIG. 2) and is less likely to be selected. (Abramson, col. 3, lines 29-33).

Even furthermore, even if one assumes that this data is based on session data, it would still be irrelevant. Claim 17 does not recite polling the session objects to obtain session data. Rather, it recites polling the session objects "to determine if they have been updated since the last time step (2) was performed". This is a totally different feature.

In the Advisory Action, the Examiner followed up on this issue, responding to the above argument by asserting (1) that "even though the session objects are not present in figure 2, it does not prove that the Examiner's assertion is not true" and (2) that "as seen in column 3, lines 24-30: when it is polling the status it is inherent that session objects would be polled as well, being that they are part of the status, which would also provide information regarding an update to the session objects".

Applicant respectfully traverses both of these points. With respect to the Examiner's first point, regardless of whether Figure 2 *per se* proves that the session data is not polled, it is not incumbent upon Applicant to establish a negative. Rather, it is the Examiner's burden to establish that Abramson discloses polling session data. As noted above, not only does Abramson not disclose it, but the available evidence suggests the opposite.

As to the second point, Applicant has already addressed this issue. Particularly, as noted above, claim 17 does not recite polling the session objects to obtain session data, which is what the Examiner is asserting is disclosed by Abramson. Rather, claim 17 recites polling the session objects "to determine if they have been updated since the last time step (2) was performed". This is a totally different recitation than what the Examiner is asserting is taught by Abramson.

Specifically, the Examiner argues that polling the status of the session data inherently includes information regarding an update to the session objects. However, this is not inherent at all. Claim 17 recites "polling said session objects stored in said memories local to said JVMs to determine if they have been updated since the last time step (2) was performed". Thus, claim 17 does not just recite polling the session objects, but then also determining if they have been updated since the last time. Simply polling a session object does not disclose if the session object has been updated since it was last written to the central memory (as claimed in claim 17). There is additional information that must be considered to make this determination, such as when the session object was last written to the central memory. There is no basis in Abramson to support an assumption that information as to the last time a session object was written

to a central memory is contained in the session object. In fact, it not only is not inherent in Abramson, but is extremely unlikely. There is nothing to indicate that it would even be possible in Abramson. In any event, at a minimum, there certainly is nothing in the Abramson reference that hints at “determin[ing] if they have been updated since the last time step (2) was performed”.

C. Claims 1-10, 15 and 18-24

The Examiner further rejected claims 1-10, 15, 18, 22, and 23 under 35 USC section 103(a) as unpatentable over Abramson, Feit, and Prabandham and further in view of Ng.

Similarly to claim 11, independent claim 1 recites “a second computer program adapted to write to said database a copy of said HTTP session data for each said HTTP session at a designated time that is a function of a predetermined time interval since the last write to said database of http session object data for said http session”.

Additionally, note that the database is previously introduced in claim 1 as being in a memory “accessible by each of said JVMs”, a limitation that clearly is not met by Feit.

Ng does not disclose the teachings lacking from the other references discussed above.

Accordingly, independent claim 1 distinguishes over the prior art for at least all of the same reasons given above in connection with independent claim 11.

Since claims 2-10 depend from independent claim 1, they distinguish over the prior art of record for at least all of the same reasons given above for claim 1.

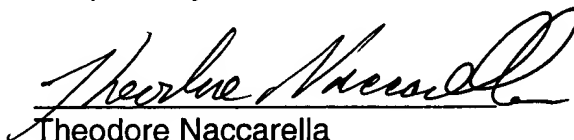
Likewise, independent claim 18 recites "a second computer program adapted to write a copy of said http session data for each said http session in said database at designated times, said designated times determined as a function of at least one of (a) the number of times the http session object data is updated in said local memory and (b) the number of times said http request in said http session is serviced". Again, note that the "database" is previously introduced in claim 18 as being in a memory "accessible by each of said JVMs".

Accordingly, independent claim 18 distinguishes over the prior art for essentially the same reasons as independent claims 1 and 11 as discussed above.

Since claims 19-24 depend from independent claim 18, they distinguish over the prior art of record for at least all of the same reasons given above for claim 1.

Respectfully submitted,

Dated: 6.22.05



Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

M:\TNACCARELLA\CLIENTS\IBM\24881\PATENT OFFICE\APPEAL BRIEF (DAS).DOC

APPENDIX A: CLAIMS INVOLVED IN THIS APPEAL

1. A server system utilizing an HttpSession object in a Java servlet application program interface (API) comprising:
 - a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;
 - a second memory having a database for storing HttpSession objects for http sessions being handled by said JVMs, said memory being accessible by each of said JVMs;
 - a first computer program adapted to store in a memory local to said server running said JVM HttpSession data for each http session handled by said JVM;
 - a second computer program adapted to write to said database a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session.
2. The server system of claim 1 further comprising a third computer program adapted to write to said database a copy of said HttpSession object data for each said http session at the time the http session is initiated.
3. The server system of claim 2 wherein said plurality of JVMs are running on a plurality of servers.
4. The server system of claim 3 wherein said writes to said database are performed at the end of a corresponding servlet service method.

5. The server system of claim 4 wherein said server system services the World Wide Web.

6. The server system of claim 5 wherein said Java servlet APIs are J2EE servlet APIs.

7. The server system of claim 1 wherein said second program polls said session objects stored in said memories local to said JVMs to determine if said predetermined time interval has passed since they have been updated and wherein said second program is adapted to write to said database only copies of said HttpSession objects that have been updated within said predetermined time interval.

8. The server system of claim 7 wherein said second computer program is invoked at predetermined intervals.

9. The server system of claim 1 wherein said time interval is configurable.

10. The server system of claim 9 wherein said time interval is between ten seconds and five minutes.

11. A method of maintaining session data in a server system servicing a network, said server system maintaining state data pertaining to sessions, said method comprising the steps of:

(1) storing data for each session in a memory local to a server servicing said session;

(2) writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions.

12. The method of claim 11 further comprising the step of:

(3) writing in said database a copy of said session data for each said http session at the time the http session is initiated.

13. The method of claim 11 wherein said server system services the World Wide Web.

14. The method of claim 11 wherein said server system comprises a plurality of Java Virtual Machines (JVMs) running on a plurality of servers, and wherein said data for said sessions comprises an HttpSession object of a Java servlet application program interface (API).

15. The method of claim 14 wherein said Java servlet APIs are J2EE servlet APIs.

16. The method of claim 11 wherein said time interval is configurable.

17. The method of claim 14 further comprising the step of:

(4) polling said session objects stored in said memories local to said JVMs to determine if they have been updated since the last time step (2) was performed; and

wherein, in step (2), only copies of said HttpSession objects that have been updated within said predetermined time interval are written to said database.

18. A server system utilizing HttpSession objects in a Java servlet application program interface (API) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;

a memory having a database for storing HttpSession objects for http sessions being handled by said plurality of JVMs, said memory being accessible by each of said JVMs;

a first computer program adapted to store in a memory local to said server running said JVM HttpSession object data for each http session handled by a JVM;

a second computer program adapted to write a copy of said HttpSession data for each said http session in said database at designated times, said designated times determined as a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced.

19. The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X HttpSession updates in said local memory, where X is an integer greater than or equal to 2.

20. The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X http requests in said http sessions, where X is an integer greater than or equal to 2.

21. The server system of claim 18 further comprising a third computer program adapted to store in said database a copy of said HttpSession object data for each said http session at the time the http session is created.

22. The server system of claim 21 wherein said plurality of JVMs are running on a plurality of servers.

23. The server system of claim 22 wherein said Java servlet APIs are J2EE servlet APIs.

24. The server system of claim 18 wherein said writes to said database are performed at the end of a first servlet service method of a corresponding http session received after said designated time.